E7.6-10144
CR-146153

RSI-SDSU-74-10

BOUNDARY-DETECTION ALGORITHM FOR LOCATING EDGES IN DIGITAL IMAGERY

(E76-10144) BOUNDARY-DETECTION ALGORITHM
FOR LOCATING EDGES IN DIGITAL IMAGERY (South
Dakota State Univ.) 32 p HC $4.00 CSCL 08M
N76-17453
Unclas
G3/43 00144

Remote Sensing Institute
South Dakota State University
Brookings, South Dakota 57006

RSI-SDSU-74-10

BOUNDARY-DETECTION ALGORITHM FOR LOCATING EDGES IN DIGITAL IMAGERY

Prepared for:

National Aeronautics and Space Administration
Johnson Space Flight Center
Houston, Texas 77053

Contract No. NAS 9-13337

by

Russell, M.J., D.G. Moore,
G.D. Nelson, and V.I. Myers

Original photography may be purchased from:
EROS Data Center
10th and Dakota Avenue
Sioux Falls, SD 57198

ADVANCED Rept. Significant Results

Remote Sensing Institute
South Dakota State University
Brookings, South Dakota 57006

Organization:	Remote Sensing Institute South Dakota State University Brookings, South Dakota 57006
Report Title:	Boundary-Detection Algorithm for Locating Edges in Digital Imagery
Report Type:	Advanced Report of Significant Results
EREP Investigation Number:	S452
NASA Contract Number:	NAS 9-13337
Authors:	Russell, M.J., D.G. Moore, G.D. Nelson, and V.I. Myers
Principal Investigator:	V.I. Myers
Date Submitted:	April 1975
NASA Technical Monitor:	Ryborn Kirby Operations Room Code TF6 Johnson Space Center Houston, Texas 77058

ABSTRACT

The initial development of a computer program which implements a boundary-detection algorithm to detect edges in digital images is described. Output products include symbol maps for the original data, for the Roberts' gradient of the data, and for the boundaries detected in the data. Histograms of the original data values, the data values of the boundary points, and the data values not at boundary points are an output. An initial evaluation of the boundary-detection algorithm was conducted to locate boundaries of lakes from LANDSAT-1 imagery. The accuracy of the boundary-detection algorithm was determined by comparing the area within boundaries of lakes located using digitized LANDSAT imagery with the area of the same lakes planimetered from imagery collected from an aircraft platform. Relative CPU time costs per data point are presented for using the computer program on an IBM 370/145 with an OS/VS operating system.

TABLE OF CONTENTS

LIST OF FIGURES

Boundary-Detection Algorithm for Locating Edges in Digital Imagery

M.J. Russell, D.G. Moore, G.D. Nelson, and V.I. Myers*

INTRODUCTION

The sensors aboard the SKYLAB workshop and other space-altitude platforms acquire synoptic data for many resource applications. An example use is for irrigation scheduling. Since irrigation scheduling is based upon field variations, a method to detect the spatial limits or edges of fields is required to conduct a field-by-field analysis using digital data. After definition of the field edges, data means and variances within fields can be determined.

A boundary-detection algorithm was developed to locate field boundaries. The procedures include: 1) data enhancement by computation of Roberts' gradient values, 2) location of data regions near boundaries by applying a threshold to Roberts' gradient data, 3) mapping of potential boundaries in Roberts' gradient data by location of local maximums, and 4) elimination of local maximums which are associated with noise or minor edges by applying a gradient-difference threshold. A preliminary single feature boundary-detection algorithm is described and illustrated.

DESCRIPTION OF THE BOUNDARY DETECTION PROCEDURE

Statement of the Problem

The radiance differences among individual ground targets are measured and recorded by the SKYLAB S-192 multispectral scanner. Many of the ground targets are relatively homogeneous clusters of radiance values in the data, e.g. a field of corn, a lake, or a fallow field. Boundaries between homogeneous clusters are located where abrupt changes in magnitudes of the radiance values occur. Therefore, a mathematical operator which measures the amount of change in data values would be helpful to locate boundaries that separate data classes.

---

* Staff Specialist, Remote Sensing Institute; Soil Scientist, Remote Sensing Institute; Associate Professor of Electrical Engineering; and Director, Remote Sensing Institute; respectively.

One Dimensional Model

An ideal boundary between two data clusters occurs at a step change in the data. Ideal data containing no boundaries have small constant gradients or have no gradients. Distortion in the form of a point spread function and some random noise added to this ideal model would form a more realistic model of actual data.

An ideal distortionless and noiseless step boundary along with its first and second derivatives is illustrated using solid lines in Figs. 1a, 1b, and 1c, respectively (page 11). There is a constant gradient within both classes of data and a step change at the boundary. The first derivative at the boundary between the two classes is an impulse function and elsewhere it is constant. The second derivative is a doublet which changes at the boundary from an infinite positive value to zero and then to an infinite negative value. The function is zero elsewhere.

A realistic form for a boundary between two data classes is shown with dots in Fig. 1. The infinite slope of the ideal step change becomes a gradual smooth change from one class to the other at the boundary. This blurring of the ideal data which produces a gradual change at the boundary is analogous to low-pass filtering or averaging of the ideal data. The first derivative of the realistic data for a boundary between two data classes is a maximum value at the boundary between the two classes (dots in Fig. 1b). The second derivative goes from a large positive value on the left side of the boundary to zero at the boundary to a large negative value on the right side of the boundary. In summary, first derivatives are infinite at ideal step boundaries and are large at boundaries in real data. The first derivatives, if no boundaries exist in noiseless data, are small constant values or zero. Therefore, a simple decision rule for location boundaries in noiseless data is to locate all points in the first derivative data that are larger than the surrounding points.

When realistic data are used for analysis, small changes in data values (noise) are normally present within the relatively homogeneous data classes. These fluctuations may be caused by actual changes in reflectance or emittance or may be caused by electronic noise introduced during data acquisition and/or manipulation. The absolute value of the first derivative will have non-zero values at actual data-value fluctuations.

Boundaries in noiseless first derivative data are located at local maximum first derivative values. A local maximum occurs in the first derivative data if a data point has a higher derivative value than adjacent points. The first derivative can reach a maximum not only at major boundaries but at locations of noise or minor boundaries. Noise locations or minor boundaries would be misclassified as edges using the definition that all local-maximum first derivatives are boundaries. The absolute value of the first derivative at a local-maximum-point is less for a minor boundary or noise than for a major boundary. Therefore, only first derivatives above a certain level or threshold can be used for initial boundary discrimination. Figure 2 illustrates two sets of one dimensional data: a boundary in noiseless data and a boundary in noisy data. Maximums one, two, and four in Fig. 2b are representative locations of minor boundaries located within each of the two relatively homogeneous data classes. The smaller maximums within the first derivative data that might be misclassified as major boundaries can be discarded. First derivative values for maximum one, two, and four are eliminated as major boundaries because they are less than the selected gradient threshold indicated in Fig. 2b.

Two Dimensional Model

With two-dimensional data, as in one-dimensional data, an abrupt change in the magnitude of the data occurs across a boundary. The first partial derivative in the direction perpendicular to the boundary measures this change. A spatial gradient can be used to measure magnitude changes in data. This value is the maximum rate of data change, regardless of direction, for a specific data location in the matrix of data. If the point is located at a boundary, the direction of maximum change will be perpendicular to the boundary.

For continuous functions F(X,Y) the magnitude of the spatial gradient $|\nabla F(X,Y)|$ is defined as:

$$|\nabla F(X,Y)| = \left[\frac{\partial F}{\partial x}^{2} + \frac{\partial F}{\partial y}^{2} \right]^{\frac{1}{2}}$$

The partial derivatives in this expression can be approximated by the first differences between data values in two orthogonal directions. A

discrete approximation of this magnitude of a spatial gradient is defined as:

$$|\nabla f_{i,j}| = \left[(F_{i,j} - F_{i+1,j+1})^2 + (F_{i,j+1} - F_{i+1,j})^2\right]^{\frac{1}{2}}$$

This discrete form is approximated by using the Roberts' gradient which is defined as:

$$|\nabla F_{i,j}| = |F_{i,j} - F_{i+1,j+1}| + |F_{i,j+1} - F_{i+1,j}|$$

Abrupt changes in the original data values which represent boundaries become relative maximums in the Roberts' gradient data. Changes due to minor variations of data values or noise within the relatively homogeneous data classes can also become maximums. However, the magnitude of these maximums is usually much smaller than those located near boundaries. A gradient threshold eliminates the smaller gradients that might become maxmums.

The use of a gradient threshold reduces computer time necessary for locating boundaries in the gradient values below this threshold. To locate a potential vertical boundary, the Roberts' gradient values, which are greater than the gradient threshold, are processed to detect if they are local maximums in the horizontal lines. A horizontal boundary is located by evaluating the Roberts' gradient values in the vertical lines in a similar manner. However, all maximums aren't boundaries. They are boundaries only if the local maximums are found while scanning the Roberts' gradient values in the gradient vector direction (normal to real boundaries). Without knowledge of the gradient vector direction, scanning the Roberts' gradient data for local maximums must procede in two orthogonal directions. Thus, local maximums are possible both along and perpendicular to the gradient vector direction. The perpendicular case generates local maximums parallel to the boundary, i.e. within a homogeneous data class.

A second derivative operation is used to eliminate the local maximums which are parallel to the boundary. The magnitude of the second derivative will vary as a function of the direction traversed while crossing a boundary. In one dimension, the second derivative of an ideal step boundary is a doublet. In two dimensions differencing the gradient values (second derivative approximation) along the gradient vector direction

produces large magnitudes on either side of the local gradient maximum. Differencing the gradient values along directions other than the gradient vector direction will result in lower magnitudes on both sides of the local gradient maximum. Therefore, a gradient-difference threshold eliminates maximums located in scans of gradient values perpendicular to the gradient vector direction (parallel to the boundary).

In summary, the boundary-detection algorithm does the following: 1) calculates the Roberts' gradient enhancing the boundaries in the original data, 2) thresholds gradient values to reduce the gradient data to only those locations near boundaries, 3) locates the relative maximums within the Roberts' gradient values greater than the gradient threshold, and 4) eliminates local maximums occurring parallel to actual edges by application of a gradient-difference threshold.

USAGE OF THE BOUNDARY-DETECTION ALGORITHM

The algorithm has been written as a Fortran computer program for an IBM 370/145 computer with an OS/VS operating system. The program is listed in the Appendix. This section describes data formats, sequences of steps, and output products. The effect of the gradient and gradient-difference thresholds is illustrated for several choices of each threshold.

The program accepts positive integer data ranging in value from zero to 255. The data can be in either one-byte words or four-byte words. The data are read line-by-line from a tape and stored on a disc. Any portion of the original matrix of data can be used by specifying the starting coordinates and the size of the data block.

The program supplies a user with 1) a symbol map with the original data quantized to 16 levels, 2) a symbol map of the Roberts' gradient data, 3) a symbol map of the boundaries, and 4) histograms. The first output is a map of the original data values. The 256 possible data values are quantized into 16 equal levels, each represented by a different symbol. An illustration of this map is provided in Fig. 3. The original data was generated by digitizing LANDSAT-1 band 7 (reflected infrared -0.8→1.1 μm) "9 1/2-inch" transparencies via the SADE system at the Remote Sensing Institute. The boundary sought was the shoreline of the lake.

A second output is a map of Roberts' gradient data. The first symbol in the symbol list of the program represents the Roberts' gradient value which is equal to the gradient threshold chosen. The symbol used for a Roberts' gradient value of the gradient threshold plus one is the second symbol. This continues until 48 symbols are used. All gradient values greater than the gradient threshold plus 47 are labeled by the 48th symbol. Examples of Roberts' gradient output maps for two different levels of gradient thresholds are illustrated in Fig. 4.

The maps illustrated in Fig. 5 were the result of applying two different gradient-difference thresholds to the remaining gradient values after the gradient threshold of 16 was applied (Fig. 4b). Note that the boundary in Fig. 5b, is not continuous. As a result, the gradient difference threshold was reduced to four. The resultant boundary is presented in Fig. 6. The symbols, presented only at boundary locations, represent the original data values. For this illustration, the symbols relate to film density.

The final output is a listing of three histograms. One histogram prints the number of data points at each value between zero to 255. The histogram of original data values at boundary points and the histogram of the data excluding those points located at boundaries are printed as separate listings. Examples of these outputs are not contained in this document.

The gradient and gradient-difference thresholds must be chosen by the user. The number of data points printed on the Roberts' gradient symbol map is dependent on the level chosen for the gradient threshold. All Roberts' gradient values of the data which are less than the gradient threshold will be represented as blank areas as illustrated in Fig. 4. The algorithm only operates on gradient values which are greater than the gradient threshold. If the gradient threshold is set too low, as illustrated in Fig. 4a, more data points must be checked for boundaries than if the threshold were set higher as in Fig. 4b. When the proper gradient threshold is selected, only data points near those boundaries of interest will be represented on the map. The ideal choice of the

gradient threshold is a threshold which eliminates as many points as possible from the set of potential boundary points, but which does not eliminate any part of the actual boundary. At this point of development, this decision must be provided by the investigator who is aided by the computer.

In summary, the procedure for using this algorithm is:

1. Print a map of the original data.
2. Print a map (either a subset containing the boundary or the total matrix) of Roberts' gradient values using a gradient threshold no larger than 5% of the potential data-value range and a gradient-difference threshold at 25% of the gradient threshold.
3. Determine the largest Roberts' gradient value which can be eliminated while retaining a continuous boundary and use this value for the final choice of gradient threshold for the entire scene.
4. Set the gradient-difference threshold at 25% of the gradient threshold chosen in Step 3 and print a new map of Roberts' gradient values.
5. If the boundary is not continuous, decrease the value of the gradient-difference threshold.
6. If the boundary is continuous and contains additional data points, increase the gradient-difference threshold.

Experience with data containing similar contrast ratios and noise levels will provide the user with a "first guess" for setting thresholds which are very close to the required threshold values. The data illustrated in this report are distributed between one and 255. The gradient threshold most often chosen was 16, and the gradient-difference threshold most often chosen was four. The boundary map for these threshold values is presented in Fig. 6.

EVALUATION OF THE ALGORITHM

Because SKYLAB S-192 data were not available during the development phase of this activity, LANDSAT-1 band-7 imagery was chosen to evaluate the algorithm. On LANDSAT-1 band-7 imagery, a high tonal contrast exists between water and the surrounding landscape. Therefore, the evaluation was completed using this special class of water boundaries. The digital data were acquired by digitization of the 1:1,000,000 scale transparencies to obtain spatially correct matrices (in contrast to using the digital tape products which are not spatially correct). The LANDSAT-1 imagery was digitized at approximately 36 data points per millimeter of film. This corresponds to approximately 27.7 m on the land surface. These data are similar in resolution to the anticipated data products from the S-192 scanner aboard SKYLAB. A gradient threshold of 16 and gradient-difference threshold of four were required to use the algorithm for these data. The sequence of maps which resulted after the various steps of the algorithm are presented in Figs. 3 through 6.

The boundary-detection algorithm was evaluated with comparisons of water areas measured on the digitized LANDSAT-1 data and areas measured by planimetering enlargement prints from Zeiss (12-"inch" focal length) photographic imagery taken from a RB-57 aircraft. The boundary determined from the digital LANDSAT-1 data was superimposed on Zeiss imagery for one of the lakes studied in Fig. 7. This boundary was within 27.7 m or one data sampling interval of the actual boundary. The area of Lake Goldsmith of Brookings County, South Dakota, by planimetering from the Zeiss imagery was 120 hectares of 297 acres. The white line superimposed onto the aircraft image is the boundary located, not from the aircraft image, in the LANDSAT-1 data. The area determined using the LANDSAT-1 data was 119 hectares of 295 acres.

The data values located at the boundary between the lake and land using these digital LANDSAT-1 data were distributed between values of 112 to 192 (Fig. 8). Therefore, a significant improvement over simple thresholding to locate the water was possible with boundary detection because no one threshold of data values would have located the same boundary that was obtained using the boundary-detection algorithm. These results imply

that for the data used, simple thresholding of the data would not locate the boundary found between land and water in the digital LANDSAT-1 data.

EXPECTED ERRORS IN COMPUTING AREAS OF WATER BODIES WHEN BOUNDARIES ARE LOCATED USING THIS PROCEDURE

An accuracy evaluation was conducted by measuring lake areas on LANDSAT data using the algorithm and comparing these areas to the actual area as measured from aircraft photography. The sampling frequency using these input data from the "9 1/2-inch" transparency at 36 points per millimeter corresponds to the ground interval of 27.7 m. For the illustration in Fig. 7, the predicted boundaries from LANDSAT-1 data were located within 27.7 m of the actual boundary. Assuming the 27.7 m value to be the maximum expected error for this specific application, the error of measuring various sized water bodies can be computed and presented as in Fig. 9. The maximum expected error in percentage of total area was computed using the 27.7 m value for circular lakes. An expected error of less than 10% can be anticipated for lakes larger than approximately 3 hectares or 7.4 acres. The lake used as an example in this report was 120 hectares or 297 acres and was measured within 0.6%. The "□" are measured error values for various sized lakes which were evaluated using similar procedures as outlined.

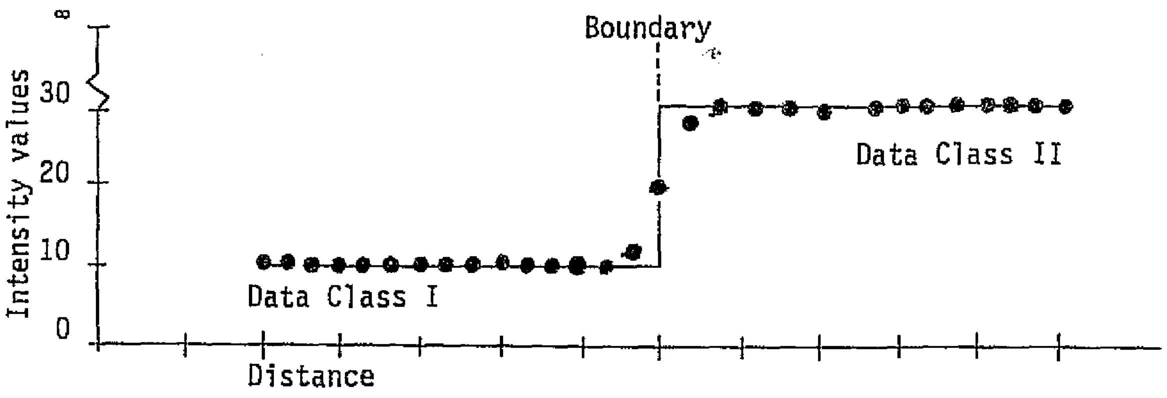
A circular lake was the least perimeter per unit area of any geometric shape. Therefore, by increasing the perimeter per unit area for irregular-shaped lakes, the error in measuring lake area increases. The maximum expected error in measuring lakes having differing shoreline development factors is illustrated in Fig. 10.

COMPUTER TIME FOR USING THE ALGORITHM

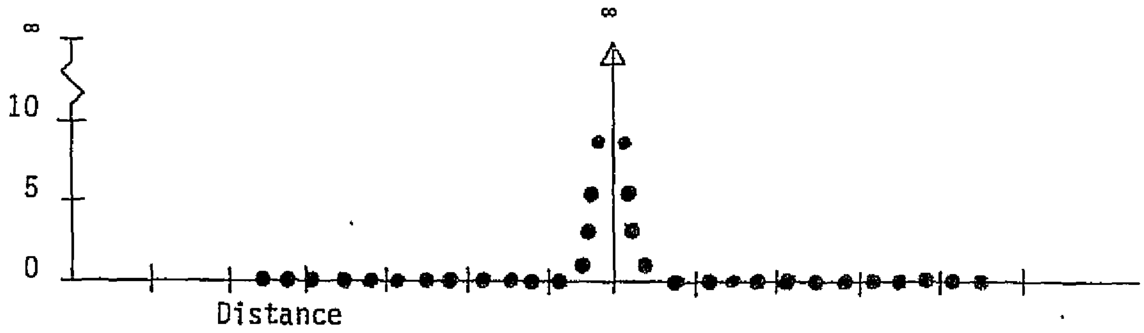
The costs for data processing are dependent upon the system used. An IBM 370/145 computer with an OS/VS operating system was evaluated. The initial compiling time for any size of data matrix was 28.8 sec. The additional computing and printing time averaged approximately 1.18 millisecond per data point. As an example, the dollar cost incurred for a 100 by 100 matrix of data (10,000 data points) would be approximately $3.40

on the South Dakota State University IBM 370/145 computer with an OS/VS operating system. This cost is based upon the initial input data as a digital tape. Figure 11 illustrates the computer time required for application of the computer program for various sizes of data matrices.

a. Plot of original data values.



b. First derivative data values of original data.



c. Second derivative data values of original data.

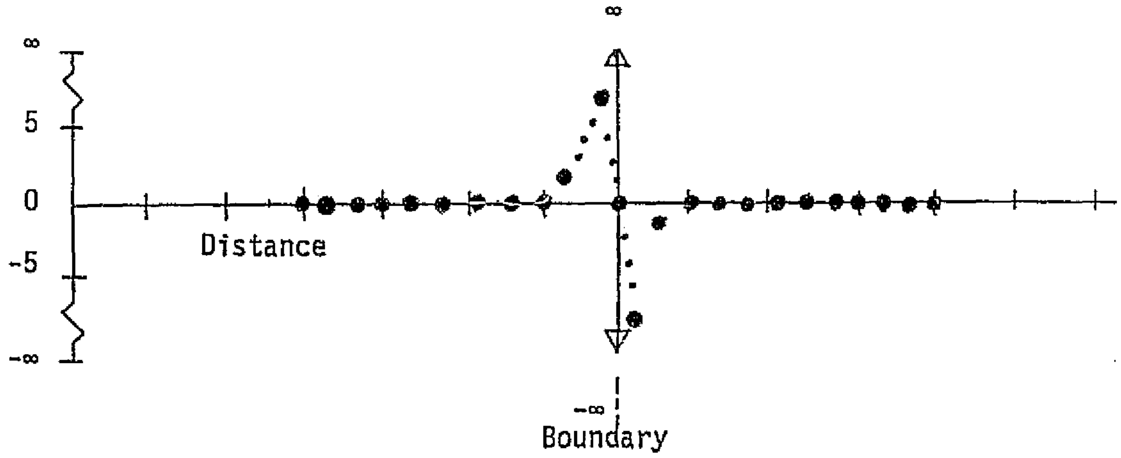
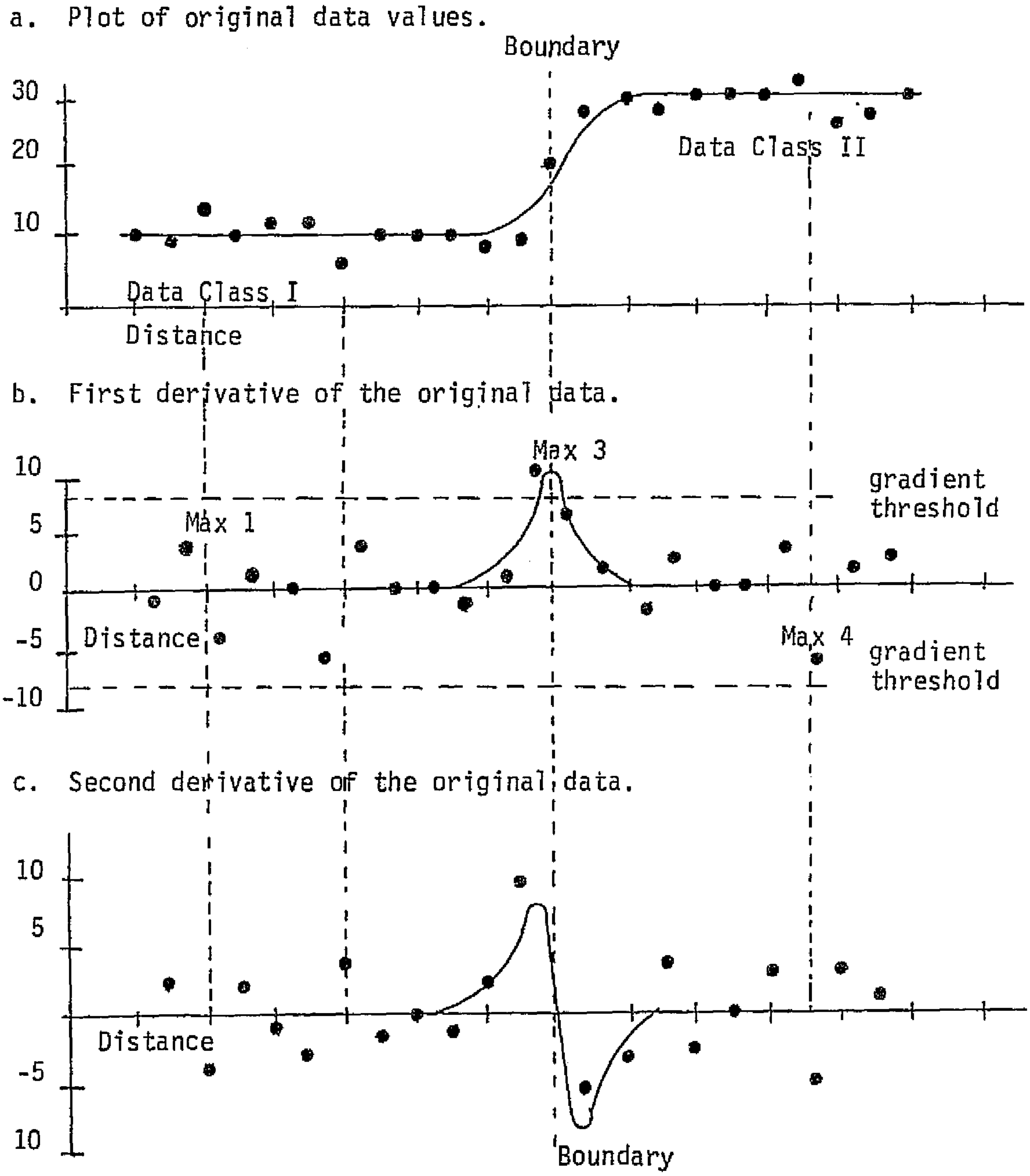


Fig. 1 - Ideal and distorted data representations for a step boundary between two data classes where the solid line represents an ideal set of data and the dots represents a distorted data set. The ideal data contains a step boundary; whereas, the data are gradually changing at the boundary.



a. Plot of original data values.

b. First derivative of the original data.

c. Second derivative of the original data.

Fig. 2 - Distorted and expected data of a step boundary where the solid line is a distorted but noiseless data set and the dots are a distorted and noisy data set.

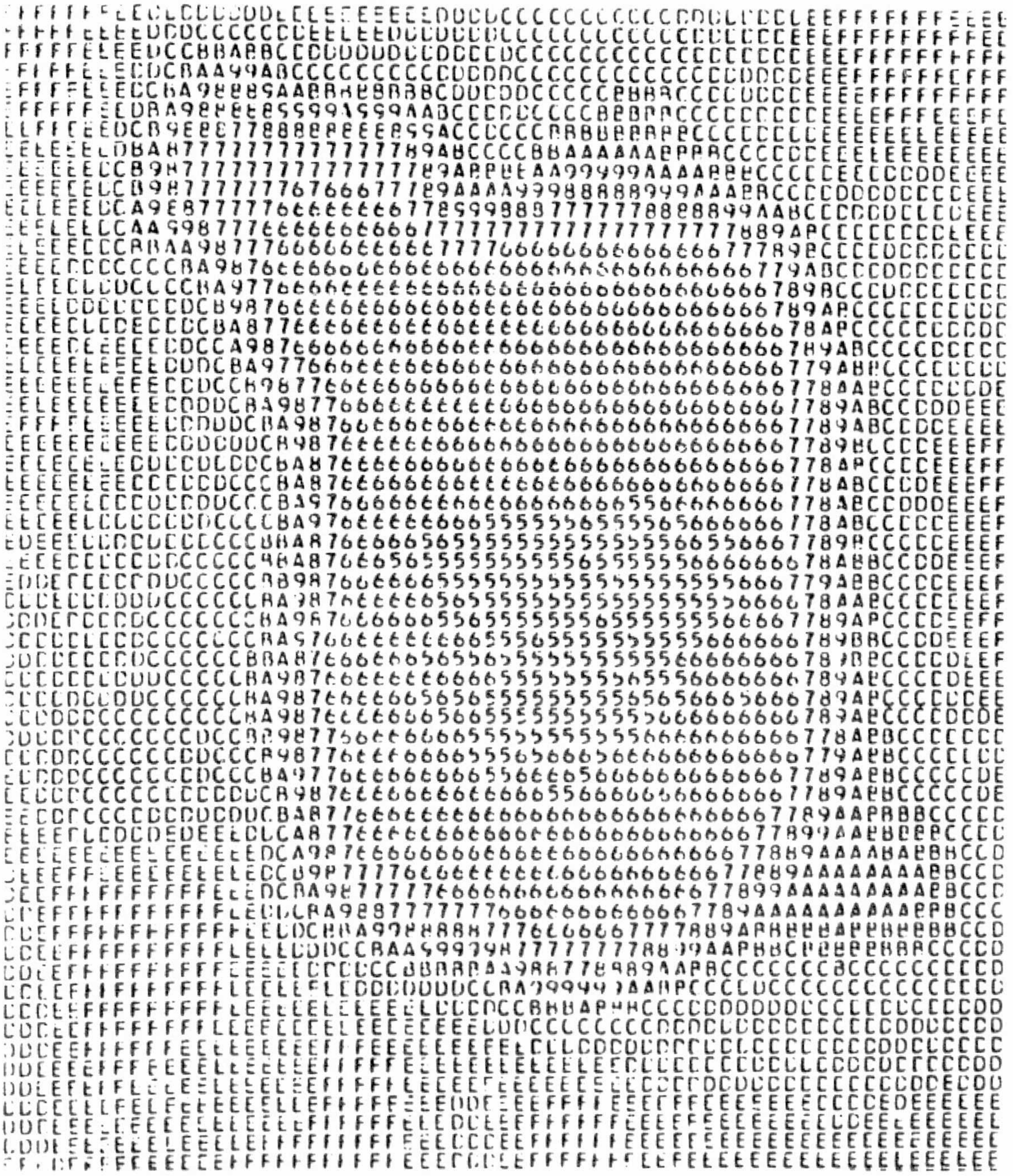
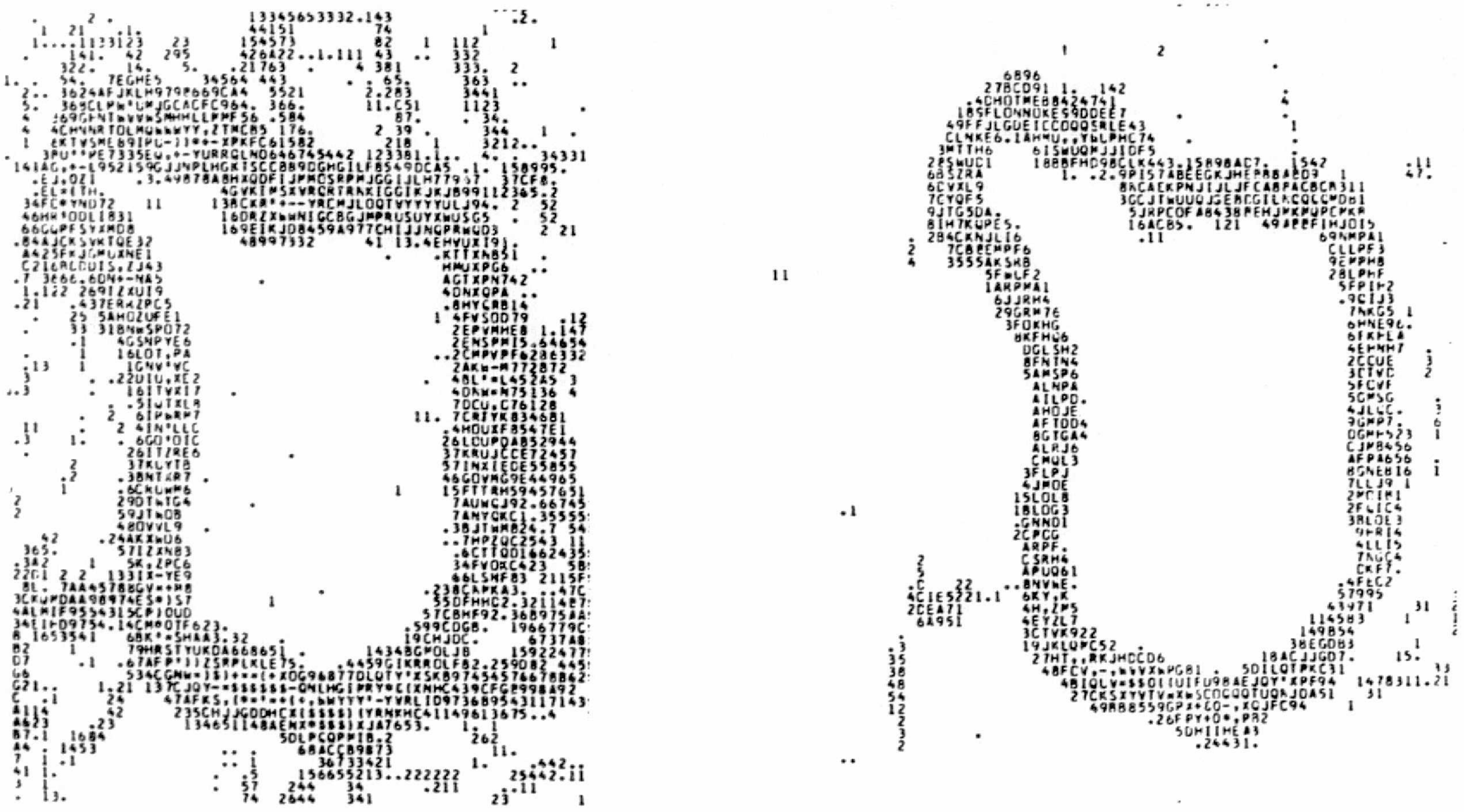


Fig. 3 - A symbol map of the original data for a digitized LANDSAT-1 image. The symbols represent film density values quantized into 16 equal levels of data which are distributed over a range of zero to 255. The 16 symbols are .,1,2,3,4,5,6,7,8, 9, A,B,C,D,E, and F, respectively. They are in order of increasing magnitude, where "." equals 0-15, ...,F equals 240-255, etc.



4a

Gradient threshold equal to 6.

4b

Gradient threshold equal to 16.

Fig. 4 - A symbol map of the Roberts' gradient of the digital LANDSAT-1 data (presented in Fig. 3) resulting from two different levels of gradient thresholds. The boundary-detection algorithm considers only these remaining points when searching for edges.

5a

Gradient-difference threshold equal to 1.

5b

Gradient-difference threshold equal to 6.

Fig. 5 - The boundary symbol map of the digital LANDSAT-1 data (presented in Fig. 3) with two choices of the gradient-difference threshold. The symbols are the same as those in the data output map (Fig. 3). The gradient threshold level was 16.

Note that in 5a the boundary is noisy, indicating that the gradient-difference threshold was too small. In 5b, the boundary lost continuity indicating the threshold was too large.

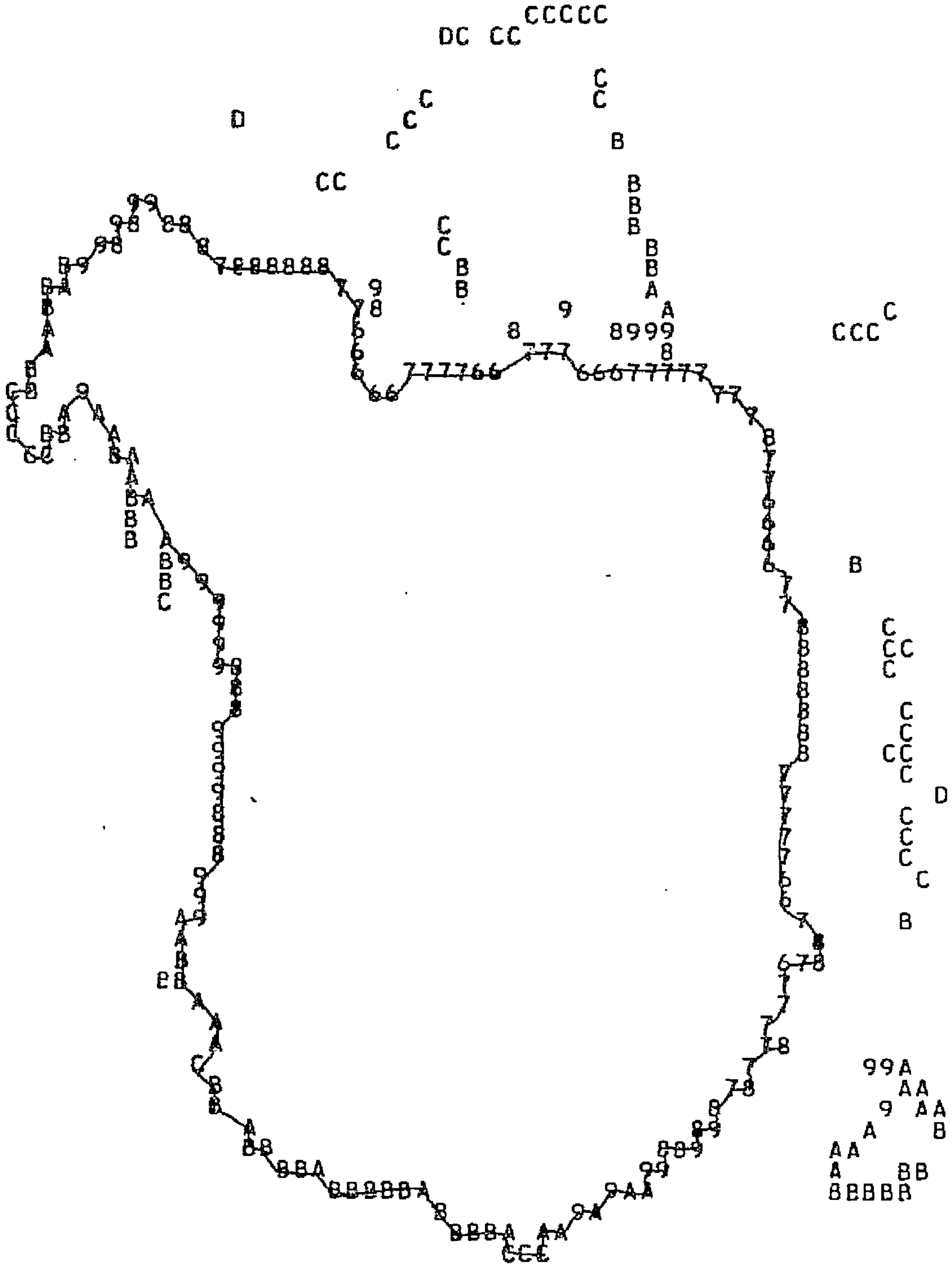


Fig. 6 - The boundary symbol map with the best choice of both gradient and gradient-difference thresholds. The gradient threshold was 16 and the gradient-difference threshold was four. An example interpretation of the boundary location is denoted by the solid line.

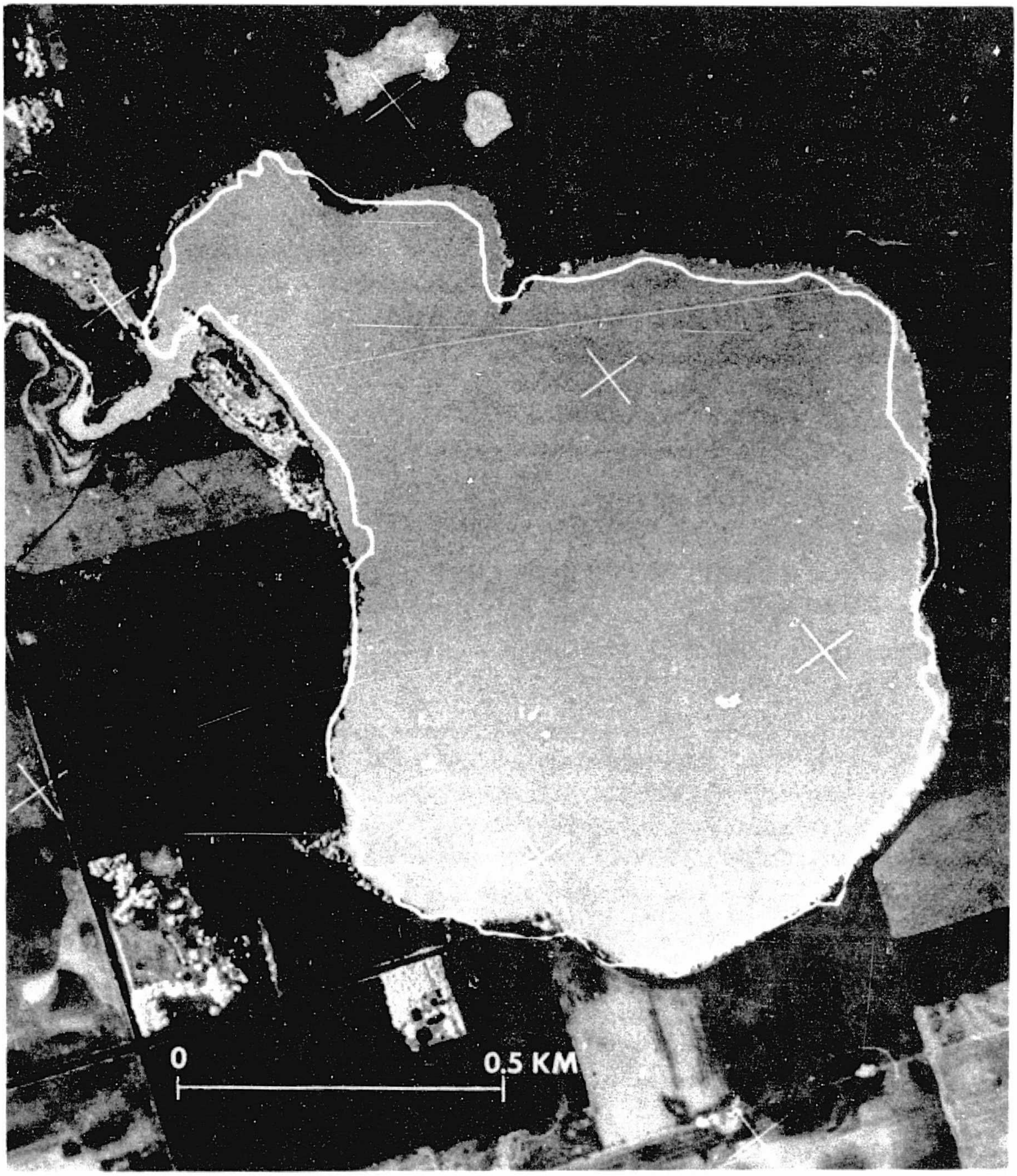


Fig. 7 - Land-lake boundaries on Zeiss imagery ("12-inch" focal length camera in RB-57 aircraft) compared with those located using digital LANDSAT-1 data. The white line is the boundary located using the boundary-detection algorithm applied to digitized LANDSAT-1 imagery taken on the same day as the aircraft image.

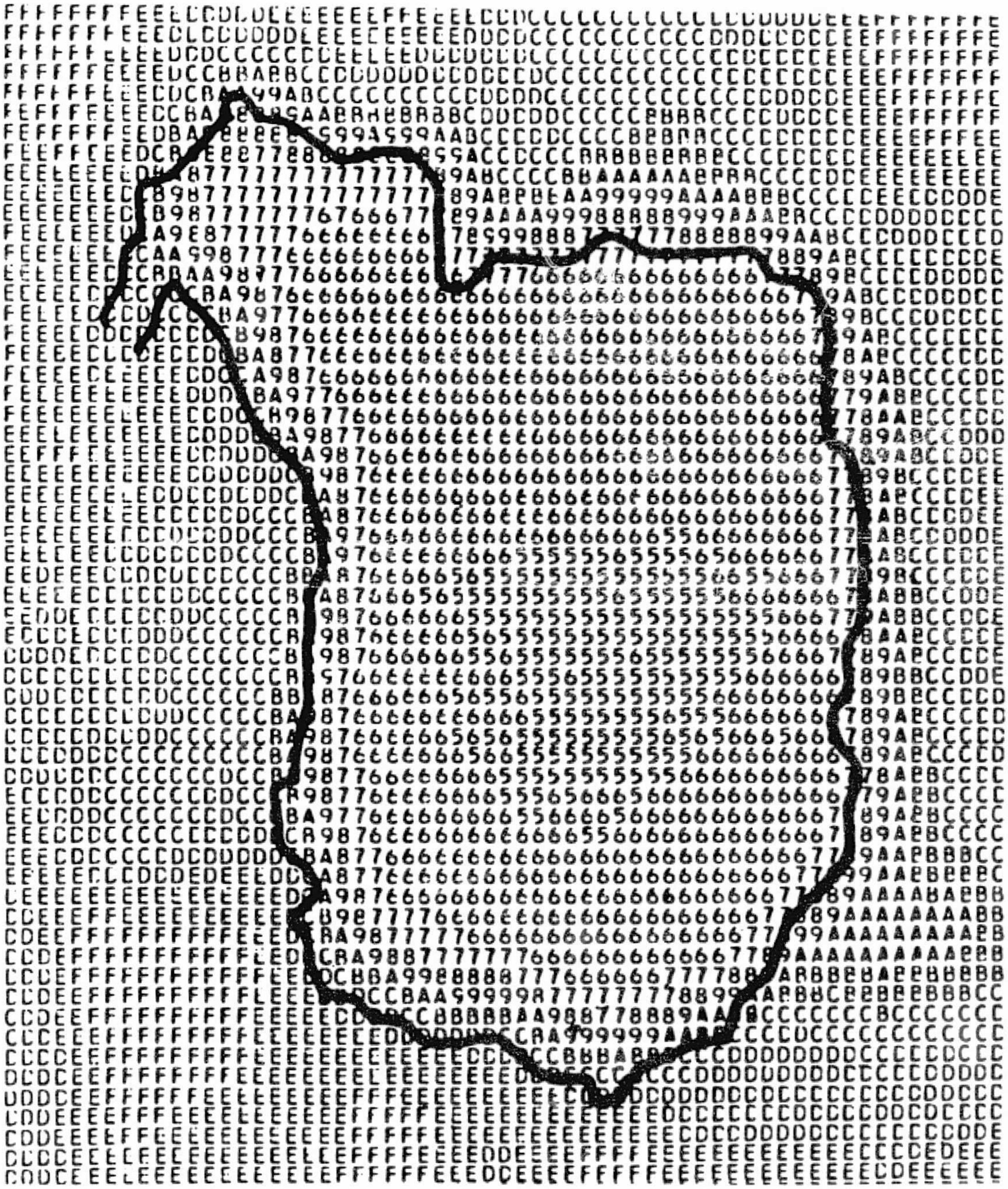


Fig. 8 - The boundary located from a digitized LANDSAT-1 image registered with a data symbol map. The symbols represent film density values quantized into 16 equal levels of data which is distributed over a range of zero to 255. Note that a simple thresholding of film density values would not have resulted in the similar boundary.

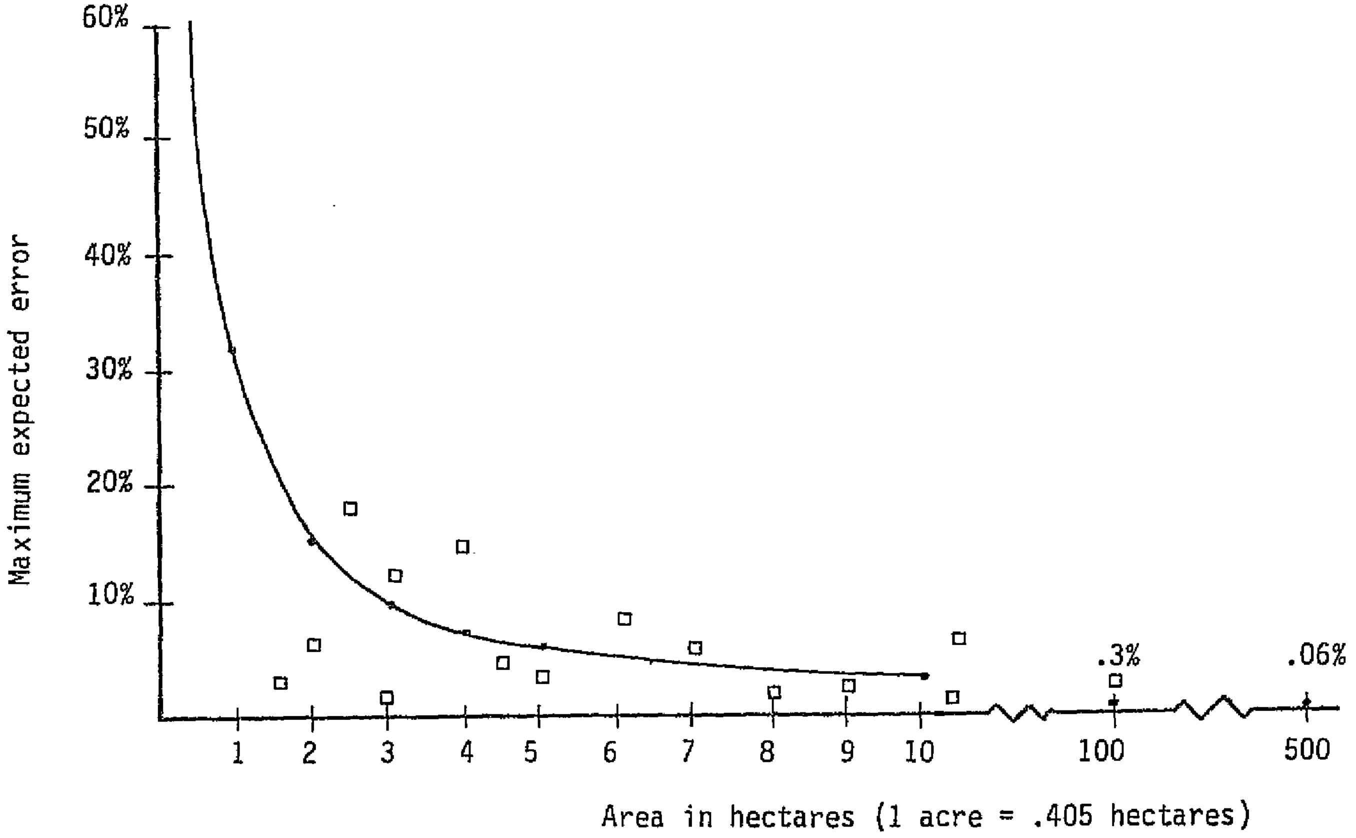


Fig. 9 - The maximum expected errors for area measurements of circular boundaries if the circle radius is overestimate by no more than 27.7 m. The "□" indicate actual error evaluations comparing processed LANDSAT-1 data to underflight Zeiss photographic data. The areas measured from the scaled aircraft enlargement prints were assumed to be accurate. The actual error for these lakes which had a shoreline development factor greater than 1 did fall within tolerable limits of the plotted maximum expected error.

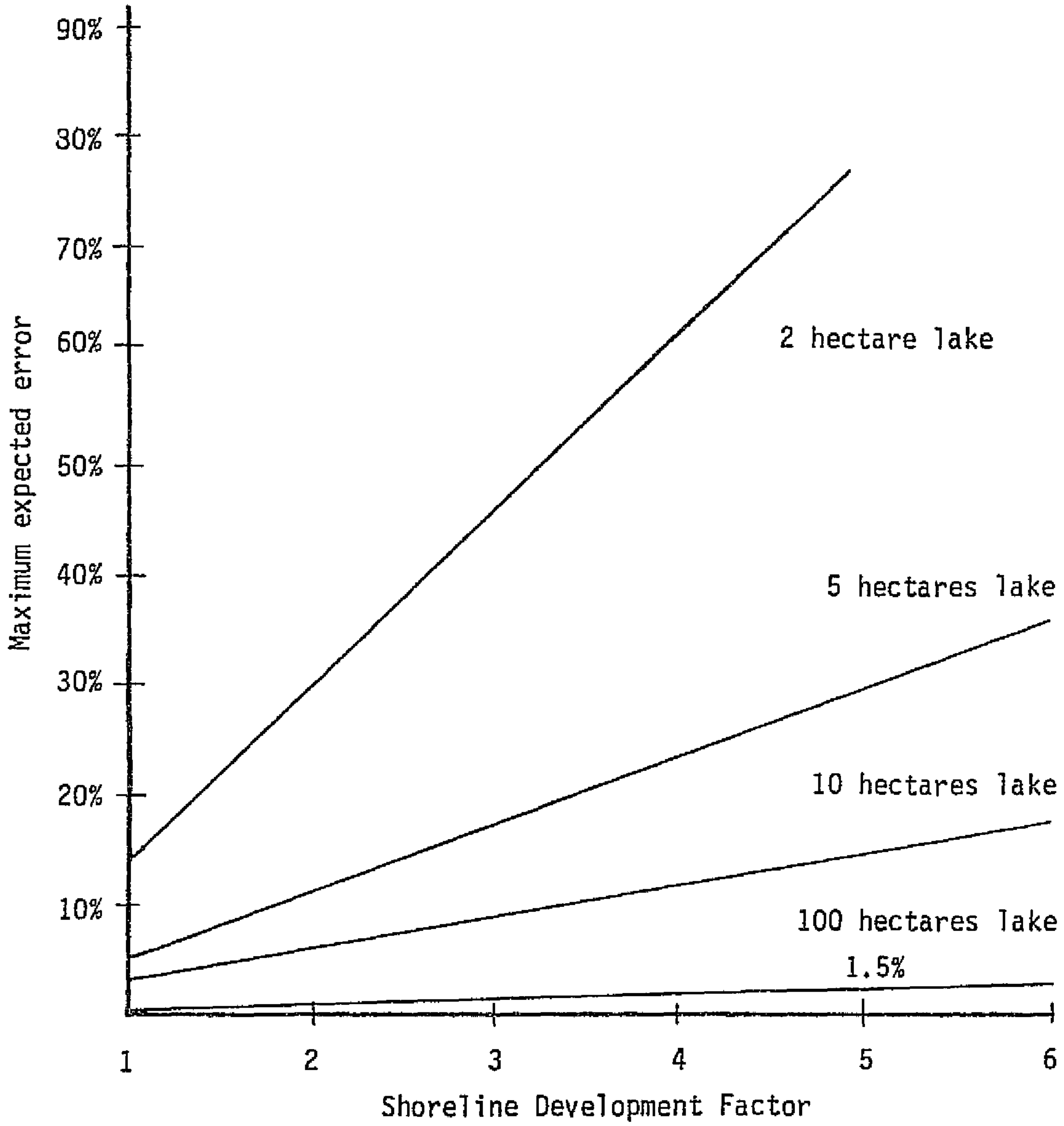


Fig. 10 - The maximum expected error for area measurements using boundaries located from digitized LANDSAT-1 images of lakes with various shoreline development factors. A lake with a shoreline development factor of one is circular and a lake with a shoreline development factor of five is quite irregularly shaped.

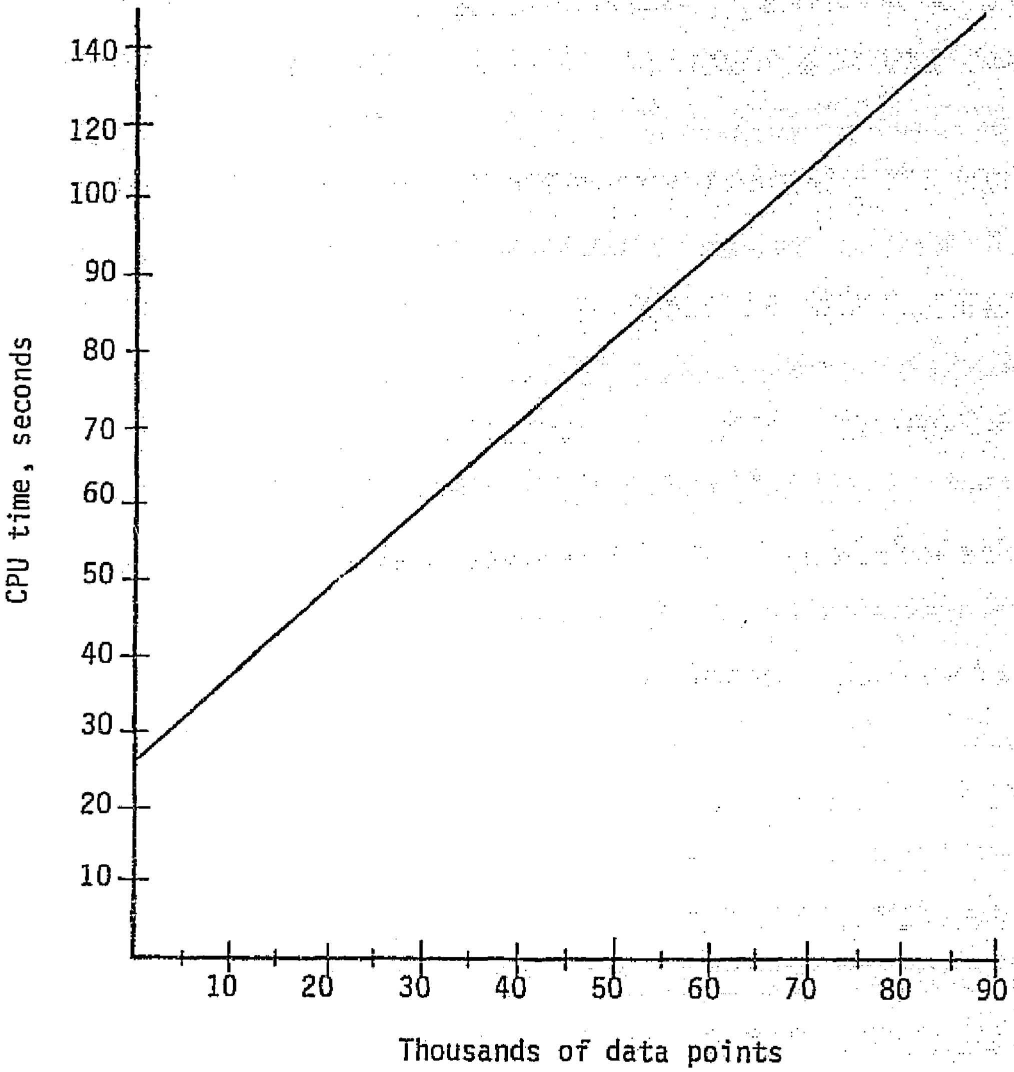


Fig. 11 - The CPU time per data point necessary for locating boundaries with an IBM 370/145 computer with an OS/VS operating system.

```
C
C                              APPENDIX
C
C
C        SAMPLE MAIN PROGRAM FOR THE DETECTION OF BOUNDARIES
C                       FROM DIGITAL DATA
C
C
C..................................................................RBND001
C..................................................................RBND005
C                                                                  RBND010
C....... PURPOSE ...................................................RBND015
C                                                                  RBND020
C     01.)READ THE INPUT MATRIX,WHICH IS DIGITIZED DATA,            RBND025
C         FROM A TAPE AND WRITE IT ON A DISC.                       RBND027
C     02.)FIND THE BOUNDARIES BETWEEN DIFFERENT CLASSES OF DATA.    RBND030
C     03.)FIND THE HISTOGRAM OF THE INPUT DATA,THE INPUT DATA MINUS RBND035
C         DATA POINTS AT  BOUNDARIES,AND THE BOUNDARY POINTS.       RBND040
C                                                                  RBND045
C                                                                  RBND050
C....... USAGE .....................................................RBND055
C                                                                  RBND060
C                                                                  RBND065
C     01.)THE DATA USED BY THIS ALGORITHM IS OBTAINED FROM MAGNETIC RBND070
C         TAPE.   A DATA WORD CAN BE IN TWO DIFFERENT FORMATS.      RBND075
C         PACKED FORMAT IMPLIES THAT 1 DATA WORD =ONE 8 BIT BYTE    RBND080
C         UNPACKED FORMAT IMPLIES THAT   1 DATA WORD = FOUR 8 BIT BYTES. RBND085
C         (THE PARAMETER IP TELLS THE PROGRAM WHICH OF THESE TWO    RBND090
C         TAPE FORMATS ARE USED)                                    RBND095
C     02.)THE PARAMETER NR, WHICH IS LISTED ON THE NEXT SHEET,      RBND100
C         MUST BE CHANGED ON THE FOLLOWING DIMENSION CARD.          RBND102
C         INTEGER*2 NDAT(NR-1,132),IRBGR(NR,132)                    RBND105
C     03.)TAPE UNIT JOB CONTROL CARDS ARE NEEDED.(UNIT=MATI=09)     RBND110
C     04.)DISC UNIT JOB CONTROL CARDS ARE NEEDED.(UNIT=8 )          RBND115
C     05.)ONE PARAMETER CARD IS USED WITH FORMAT(15I5).             RBND120
C     06.)IF THE SYMBOL SEQUENCE IS CHANGED THEN A SECOND           RBND122
C         PARAMETER CARD IS NEEDED . USE A FORMAT OF 20A4           RBND125
C         FOR 48 SYMBOLS.                                           RBND130
C     07.)IF THE DATA IS PACKED SUBROUTINE RSIUN? IS NECESSARY      RBND132
C         TO UNPACK THE DATA.                                       RBND133
C                                                                  RBND134
C                                                                  RBND135
C....... DESCRIPTION OF PARAMETERS.  ...............................RBND136
C                                                                  RBND137
C     01.)LBL  =LOGICAL BLOCK LENGTH OF THE DATA BLOCK USED.        RBND140
C         IT MUST BE DIVISIBLE BY FOUR EVENLY.                      RBND150
C     02.)NR = NUMBER OF RESOLUTIONS WANTED. THAT IS THE NUMBER     RBND160
C         OF TOTAL SCAN LINES READ FROM THE TAPE TO MAKE ONE PICTURE. RBND170
C     03.)NEX=LBL/128  IF THIS NUMBER IS NOT EVENLY DIVISIBLE,      RBND180
C         ADD ONE TO THE ANSWER NOT USING THE REMAINDER.  IF IT IS  RBND190
C         EVENLY DIVISIBLE BY 128 THEN THAT IS YOUR VALUE FOR NEX.  RBND200
C     04.)IRES = IS THE RESOLUTION AT WHICH YOU WANT TO WORK ON THE RBND210
C         DATA. (EXAMPLE..IRES=1 IMPLIES EVERY LINE EVERY POINT),   RBND220
C         (EXAMPLE..IRES=3 IMPLIES EVERY THIRD LINE, EVERY THIRD POINT). RBND230
C     05.)ICHAN= IF THE SYMBOL SEQUENCE IS TO BE CHANGED WITH ANOTHER RBND240
C         DATA CARD, A ONE IS PLACED HERE. IF A CHANGE IS NOT WANTED, RBND260
C         A ZERO IS PLACED IN THIS FIELD.                           RBND270
C     06.)IP=1 IF THE DATA IS PACKED, IP=0 IF THE DATA IS UNPACKED. RBND280
C     07.)MX IS THE X COORDINATE OF THE START OF THE DATA BLOCK USED RBND290
C         FROM THE TAPE AND LBL IS THE X COORDINATE AT THE END.     RBND300
C     08.)MY IS THE Y COORDINATE OF THE START OF THE DATA BLOCK     RBND320
C         USED FROM THE TAPE AND NR IS THE Y COORDINATE OF THE END. RBND320
C     09.)NTHS IS A THRESHOLD FOR NOISE NEAR BOUNDARIES.            RBND330
C         A TYPICAL RANGE FOR NTHS...  4 GT NTHS LT 12              RBND333
```

```
C     10.)LRBGR IS THE THRESHOLD FOR THE GRADIENT VALUE FOR NOISE WITHIN
C         DATA CLASSES NOT NEAR BOUNDARIES.
C         A TYPICAL RANGE FOR LRBGR...  6 GT LRBGR LT 24
C
C
C ...... METHOD    ...................................................
C
C     01.)READ THE TAPE OF INPUT MATRIX IM.
C     02.)WRITE THE INPUT MATRIX IM ON THE DISC.
C     03.)OPERATE ON THE FIRST 128 COLUMNS OF DATA STORED ON THE DISC.
C         128 COLUMNS WAS CHOSEN BECAUSE ONLY 128 POINTS CAN BE PRINTED
C         PER LINE PRINTER PAGE WIDTH.
C     04.)UNPACK THE DATA IF NECESSARY.
C     05.)TAKE THE ROBERTS GRADIENT OF THE DATA...
C         ROBERTS GRADIENT =IABS(NDAT(I,J)-NDAT(I+1,J+1))+
C         IABS(NDAT(I,J+1)-NDAT(I+1,J))
C     06.)WRITE A PAGE OF DATA NDAT IN SYMBOL NOTATION.
C         QUANTIZED INTO 16 LEVELS.
C     07.)WRITE A PAGE OF ROBERTS GRADIENT IN SYMBOL NOTATION .
C           SYM(1)=IS EQUIVALENT TO LRBGR
C           SYM(I)= IS EQUIVALENT TO LRBGR+I-1
C     08.)THE BOUNDARY IS FOUND FROM ROBERTS GRADIENT INFORMATION
C         DATA  IN BOTH THE X AND Y DIRECTIONS.
C     09.)A BOUNDARY CANNOT BE FOUND FROM ROBERTS GRADIENT INFORMATION-
C         IRBGR(I,J) IF THE GRADIENT IS LESS THAN THE THRESHOLD LRBGR.
C     10.)WHEN A MAXIMUM IS FOUND WHILE WORKING IN THE X OR Y DIRECTIONS
C         THE MAXIMUM DIFFERENCE IG IS CALCULATED BETWEEN THE MAXIMUM
C         AND THE GRADIENT POINTS ON EITHER SIDE OF THIS POINT
C         IN THE X OR Y DIRECTION RESPECTIVELY.
C     11.)THE THRESHOLD NTHS ELIMINATES ALL BOUNDARY POINTS IN WHICH THE
C         MAXIMUM DIFFERENCE IG AS CALCULATED ABOVE IS LESS THAN NTHS.
C     12.)THE BOUNDARY IS CLASSIFIED BY FINDING THE AVERAGE OF THE
C         FOUR DATA POINTS SURROUNDING THE BOUNDARY POINT.
C     13.)A SYMBOL PRINTOUT QUANTIZED INTO 16 SYMBOLS OF THE DATA
C         POINTS AT A BOUNDARY IS PRINTED ON THE LINE PRINTER.
C     14.)THE ABOVE PROCESS IS REPEATED IF NECESSARY ON THE SECOND 128
C         COLUMNS OF DATA.
C     15.)THE HISTOGRAM OF THE DATA, THE HISTOGRAM OF THE DATA MINUS
C         THE BOUNDARY POINTS, AND THE HISTOGRAM OF THE DATA IS
C         NOW PRINTED OUT.
C
C.....................................................................
C
C
C
C
C
      INTEGER*2 IRBGR(90,132),NDAT(90,132)
      INTEGER*2  NA(132),NB(132)
      DIMENSION SYM(48)
      DIMENSION DEC(154)
      DIMENSION KCOUNT(256)
      DIMENSION  JCOUNT(256)
      DIMENSION ICOUNT(256)
      COMMON/COMWRC/LINE(2048)/COMBYT/LINEPK(512)
      DATA SYM/'    ','....','1111','2222','3333','4444','5555',
     L'6666','7777','8888','9999','AAAA','BBBB','CCCC','DDDD','EEEE',
     6'FFFF','GGGG','HHHH','IIII','JJJJ','KKKK','LLLL','MMMM','NNNN',
     6'OOOO','PPPP','QQQQ','RRRR','SSSS','TTTT','UUUU','VVVV','WWWW',
     6'XXXX','YYYY','ZZZZ',',,,,','****','----','====','++++','((((',
     6'****','CCCO','))))','EEEE','$$$$'/
C        ICR= CARD READER AS INPUT.
C        LP= LINE PRINTER AS OUTPUT.
C        MAGT= MAGNETIC TAPE AS INPUT.
      ICR=1.
```

```
      LP=12
      MAGT=09
      READ(ICR,9)LBL,NR,NEX,IRES,ICHAN,IP,MX,MY,NTHS,LRBGR
      IF(ICHAN.EQ.1) READ(ICR,42) SYM
      LBLW=LBL/4
      NR=NR-IRES
      NRR=NR/IRES
      DO 1 IH=1,255
      KCOUNT(IH)=0
    1 ICOUNT(IH)=0
      REWIND 8
      DO 513 I=1,MY
  513 READ(MAGT,40 )(LINEPK(ID),ID=1,LBLW)
C READ TAPE WRITE TO DISC
      DO 03 I=1,NR,IRES
      DO 11 IG=1,IRES
   11 READ(MAGT,40 )(LINEPK(ID),ID=1,LBLW)
   03 WRITE(8)(LINEPK(ID),ID=1,LBLW)
      DO 8 LM=1,NEX
      WRITE(LP,10)
      L=MX+(130*IRES)
      IF(LM.GT.1.AND.L.LT.LBL)  L=L+1
      IF(LM.EQ.NEX.AND.IP.EQ.0)L=LBLW
      IF(LM.EQ.NEX.AND.IP.EQ.1)L=LBL
      JB=(L-MX)/IRES +1
      JCD=JB-1
      JKO=JCD
      IF(JCD.GT.128)  JKO=128
      REWIND 8
      WRITE(LP,304)LM
      DO 4 JJJ=1,NRR
C         UNPACK THE DATA
      READ(8)(LINEPK(ID),ID=1,LBLW)
      IF(IP.EQ.1)GO TO 6
      DO 7 JKD=1,LBLW
    7 LINE(JKD)=LINEPK(JKD)
      GO TO 12
    6 CALL RSIUNP
   12 IF(JJJ.EQ.1)  GO TO 800
      DO 21 KK=1,JB
   21 NA(KK)=NB(KK)
  800 MB=0
      DO 5 KK=MX,L,IRES
      MB=MB&1
      NDAT(JJJ,MB)=LINE(KK)
    5 NB(MB)=LINE(KK)
C ROBERTS GRADIENT
      IF(JJJ.EQ.1)      GO TO 4
      DO 22 KKK=1,JCD
      JE=KKK&1
      IYRT=NA(KKK)-NB(JE)
      IXRT=NA(JE)-NB(KKK)
      A=IABS(IYRT)&IABS(IXRT)
   22 IRBGR(JJJ,KKK)=A
C         WRITE THE DATA NDAT
      DO 41 KK=1,JKO
      NUTS=NB(KK)
      JD=(NUTS/16)+2
      IF(NUTS.EQ.0)JD=1
      DEC(KK)=SYM(JD)
      ICOUNT(NUTS)=ICOUNT(NUTS)&1
      JCOUNT(NUTS)=ICOUNT(NUTS)
   41 CONTINUE
      WRITE(LP,3)(DEC(I),I=1,JKO)
    4 CONTINUE
```

```
C          WRITE ROBERTS GRADIENT
      WRITE(LP,10)
      WRITE(LP,10)
      WRITE(LP,305)LM
      DO 25 I=2,NRR
      DO 26 J=1,JKO
      JD=IRBGR(I,J)-LRBGR
      IF(JD.GE.48)JD=48
      IF(JD.LT.1)JD=1
   26 DEC(J)=SYM(JD)
   25 WRITE(LP,2)(DEC(IM),IM=1,JCD)
      MX=IRES*LM*128-1
      JCE=JCD-1
      NRE=NRR-1
      WRITE(LP,10)
      WRITE(LP,10)
      WRITE(LP,306)LM
      DO 29 I=3,NRE
      IPLUS=I+1
      IMINS=I-1
      DO 27 J=2,JCE
      JPLUS=J+1
      JMINS=J -1
      IFD=IRBGR(I,J)
      IG=0
      II=0
      JD=1
      IF(IFD.LE.LRBGR) GO TO 250
C          CHECK FOR X BOUNDARIES
   51 IB=IRBGR(IPLUS,J)
      IBB=IRBGR(IMINS,J)
      IF(IB.GT.IFD.OR.IBB.GT.IFD)  GO TO 52
      IG   =IFD -IBB
      IB=IFD-IB
      IF(IB.GT.IG)    IG=IB
   52 CONTINUE
C CHECK FOR Y BOUNDARIES
   53 ID= IRBGR(I,JPLUS)
      IDD= IRBGR(I,JMINS)
      IF(ID.GT.IFD.OR.IDD.GT.IFD)    GO TO 54
      II=IFD-IDD
      ID=IFD-ID
      IF(ID.GT.II)   II=ID
   54 CONTINUE
      IF(II.GT.IG)   IG=II
      IF(IG.LE.NTHS)GO TO 250
      JD=NDAT(I,JMINS)+NDAT(I,J)+NDAT(IPLUS,JMINS)+NDAT(IPLUS,J)
      X=FLOAT(JD)/4.0
      JD=IFIX(X)
      JCOUNT(JD)=JCOUNT(JD)+1
      KCOUNT(JD)=KCOUNT(JD)+1
      X=X/16.0  +1
      JD=IFIX(X)
      IF(JD.LE.1)JD=1
  250 DEC(J)=SYM(JD)
   27 CONTINUE
      WRITE(LP,2)(DEC(IM),IM=2,JCE)
   29 CONTINUE
    8 CONTINUE
C          FORM A HISTOGRAM OF THE DATA NDAT
      WRITE(LP,10)
      WRITE(LP,303)
      WRITE(LP,301)
      WRITE(LP,302)
      WRITE(LP,[illegible])(J,ICOUNT(J),JCOUNT(J),KCOUNT(J),J=1,255)
```

```
    2 FORMAT(' ',129A1)
    9 FORMAT(16I5)
   10 FORMAT('1')
   20 FORMAT('0',5X,A1,5I5)
   40 FORMAT(3(20A4))
   42 FORMAT(20A4)
  100 FORMAT('0',5X,A1,I5)
  300 FORMAT('0',I4,14X,I6,10X,I6,5X,I6)
  301 FORMAT(' ',1X,6HNUMBER,2X,17HTIMES SYMBOL USED)
  302 FORMAT(' ',21X,'DATA',3X,'DATA MINUS BOUNDARY',5X,'BOUNDARY')
  303 FORMAT('0',30X,'DATA HISTOGRAM')
  101 FORMAT(3(20A4))
  304 FORMAT(' ',50X,'DATA PAGE',I2)
  305 FORMAT(' ',50X,'ROBERTS GRADIENT PAGE',I2)
  306 FORMAT(' ',50X,'BOUNDARY DATA PAGE',I2)
 1000 FORMAT(3(20A4))
      STOP
      END

*                                 SUBROUTINE  RSIUNP
*      TWO ARRAYS ARE INVOLVED IN THIS SUBROUTINE EACH OF WHICH STORES
*     2048 CHARACTERS.  'COMWRD' STORES ONE CHARACTER PER WORD (RIGHT
*      JUSTIFIED) AND 'COMBYT' STORES ONE PER BYTE.
*      THIS SUBROUTINE FIRST ZEROS COMWRD, THEN MOVES EACH BYTE FROM
*      COMBYT TO THE CORRESPONDING WORD IN COMWRD AND STORES IT RIGHT
*      JUSTIFIED.  THUS IT UNPACKS THE DATA FROM THE BYTE ARRAY AND
*      STORES IT RIGHT JUSTIFIED IN THE WORD ARRAY.
*                ENTERED BY 'CALL RSIUNP'
*      CALLING PROGRAM REQUIRES COMMON STATEMENT
*               COMMON/COMWRD/LINE(2048)/COMBYT/LINEPK(512)
*                'COMWRD' AND 'COMBYT' ARE REQUIRED NAMES
*                'LINE' AND 'LINEPK' ARE ARBITRARY BUT
*                '2048' AND '512' ARE REQUIRED
*      PROGRAM IS REUSABLE
*                                 REMOTE SENSING INSTITUTE
*                                 SOUTH DAKOTA STATE UNIVERSITY
*                                 BROOKINGS SOUTH DAKOTA 57006
*                                          BEL JUNE 1972
         PRINT NOGEN
         ENTRY RSIUNP
         EXTRN COMWRD
         EXTRN COMBYT
         USING *,15                       R15 BASE REGISTER
RSIUNP   STM        14,12,12(13)          SAVE FORTRAN REGISTER
         L          5,=A(COMWRD)          PUT ADDRESS OF FIRST WORD IN R5
         L          7,=A(COMBYT)          PUT ADDRESS OF FIRST BYTE IN R7
         L          3,=F'2048'            PUT 2048 IN R3
         L          4,=F'4'               PUT 4 IN R4
         L          6,=F'3'               PUT 3 IN R6
         L          8,=F'1'               PUT 1 IN R8
ZERO     MVC        0(4,5),=F'0'          * ZERO
         AR         5,4                        WORD
         BCT        3,ZERO                        ARRAY *
         L          3,=F'2048'            PUT 2048 IN R3
         L          5,=A(COMWRD)          PUT ADDRESS OF FIRST WORD IN R5
         AR         5,6                   ADDRESS LAST BYTE OF THE WORD
MOVE     MVC        0(1,5),0(7)           MOVE FROM BYTE ARRAY TO WORD ARRAY
         BCT        3,NEXT                HAVE ALL DATA BEEN CONVERTED
         LM         2,12,28(13)           * IF SO RETURN
         MVI        12(13),X'FF'               CONTROL TO
         BR         14                              FORTRAN PROGRAM *
NEXT     AR         5,4
         AR         7,8                   PUT ADDRESS OF NEXT BYTE IN R7
         B          MOVE
         END
```

REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR